

RALF MOLL, citizen of Germany, whose residence and post office address is Haselhofstrasse 32, 91058 Erlangen , Germany, has invented certain new and useful improvements in a

PROGRAMMING AND OPERATING METHOD FOR A
PROGRAMMABLE INDUSTRIAL CONTROLLER, IN PARTICULAR
A CNC CONTROLLER

of which the following is a complete specification:

PROGRAMMING AND OPERATING METHOD FOR A PROGRAMMABLE INDUSTRIAL CONTROLLER, IN PARTICULAR A CNC CONTROLLER

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application claims the priority of German Patent Application, Serial No. 102 49 677.3, filed October 24, 2002, pursuant to 35 U.S.C. 119(a)-(d), the disclosure of which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0002] The present invention relates to a programming method and an operating method for a programmable industrial controller, in particular a CNC controller. The present invention is also directed to a corresponding programmable industrial controller, which can also be a CNC controller.

[0003] In conventional methods, a sequence of control commands is generated, either directly or indirectly, for example with a CAD program, which can be interpreted and also executed by the controller. The actual sequence of control commands is in conventional applications stored in the controller and processed by the controller during the control operation.

[0004] The sequence of control commands is typically stored in the controller in the form of ASCII code. The sequence of control commands is therefore freely accessible and changeable. Since the control commands can be easily changed, usage limitations can either not be implemented at all or can be easily removed. The code is also quite lengthy and requires a significant amount of memory space, so that storing the ASCII code in the memory of a CNC controller takes a long time.

[0005] It would therefore be desirable and advantageous to provide a programming method and a corresponding operating method for a programmable industrial controller, in particular a CNC controller, which obviates prior art shortcomings and is able to specifically reduce data transfer times and memory requirements of a controller.

SUMMARY OF THE INVENTION

[0006] According to one aspect of the invention, a method for programming a programmable industrial controller, in particular a CNC controller, includes the steps of automatically coding into a binary code a sequence of control commands capable of being interpreted and executed by the controller, supplying the binary code to the controller, and reconstructing the original sequence of the control commands in the controller.

[0007] According to another aspect of the invention, a method for operating a programmable industrial controller, in particular a CNC controller, includes the steps of reconstructing a sequence of control commands from a binary code stored in the controller, with the controller capable of interpreting and executing the control commands, and controlling with the reconstructed sequence of control commands a controlled device, such as a machine tool.

[0008] According to yet another aspect of the invention, a computer program, residing on a computer-readable medium, includes instructions for causing a programmable industrial controller, in particular a CNC controller, to reconstruct a sequence of control commands from a binary code stored in the controller, with the controller capable of interpreting and executing the control commands, and to control with the reconstructed control commands a controlled device, such as a machine tool.

[0009] According to still another aspect of the invention, a programmable industrial controller, in particular CNC controller, which is capable of interpreting and executing control commands, includes means for reconstructing a sequence of control commands from a binary code stored in the controller, and means for controlling with the reconstructed sequence of control commands a controlled device, such as a machine tool.

[0010] The binary code is advantageously stored in the controller, whereby

the controller reconstructs the original sequence of control commands and controls a control device, for example a machine tool, immediately after the control commands have been determined or extracted. Accordingly, not all the control commands are extracted at the beginning of the control process and are not persistently stored in the controller. Instead, the control commands reside only temporarily in the control device during the operation of the controller.

[0011] The binary code can be compressed relative to the sequence of control commands and hence uses only a small amount of memory space.

[0012] If the binary code from the sequence of control commands is encrypted, then the sequence of control commands can only be decrypted with great difficulty or cannot be decrypted at all.

[0013] Several other measures can be implemented in conjunction with the encoding.

[0014] For example, the encrypted binary code can include a usage limitation which can be used to limit the number of control commands sequences that can be used. The usage limitation is tested during the operation of the controller and can include various criteria. Examples for a usage limitation are an absolute time, (an execution date), an elapsed time since the first use, a usage duration expressed in operating hours, a number of calls to the sequence

of control commands or any combination of these features.

[0015] Alternatively or in addition, the encrypted binary code can also include an individualization code which can be used to restrict the execution of the sequence of control commands to one or more specific controllers. For example, modern microprocessors include an microprocessor-specific code. This code can be queried before a sequence of control commands is executed and compared with the individualization code.

[0016] The encrypted binary code can also include a user code which can be used to limit processing of the sequence of control commands to one or more particular users, for example one specified user. For example, a password can be requested from the user and compared with the user code.

[0017] Finally, the encrypted binary code can include a comment for the sequence of control commands. The comment can include, for example, the author's name, the creation date and/or the first recipient of the binary code. In particular, storing the first recipient's name can facilitate legal action if intellectual property rights are violated or if the binary code was transmitted to a third party without authorization.

[0018] The encryption methods for creating the binary code can have many forms. Advantageously, the binary code is encrypted with a public key.

BRIEF DESCRIPTION OF THE DRAWING

[0019] Other features and advantages of the present invention will be more readily apparent upon reading the following description of currently preferred exemplified embodiments of the invention with reference to the accompanying drawing, in which:

[0020] FIG. 1 depicts schematically a system for generating a binary code; and

[0021] FIG. 2 depicts schematically a programmable industrial controller.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0022] Throughout all the Figures, same or corresponding elements are generally indicated by same reference numerals. These depicted embodiments are to be understood as illustrative of the invention and not as limiting in any way. It should also be understood that the drawings are not necessarily to scale and that the embodiments are sometimes illustrated by graphic symbols, phantom lines, diagrammatic representations and fragmentary views. In certain instances, details which are not necessary for an understanding of the present invention or which render other details difficult to perceive may have been omitted.

[0023] Turning now to the drawing, and in particular to FIG. 1, there is shown a system for generating a binary code which includes a compressor 1 and a coder 2. A sequence 3 of control commands as well as additional information 4 are supplied to the compressor 1. As seen in FIG. 1, the control commands are generated in ASCII code. The compressor 1 compresses the supplied information 3, 4 and transmits the compressed version of the code to the coder 2. The coder encrypts the supplied information by using a public key. The output signal of the coder 2 represents a binary code which is made available to a programmable industrial controller 5 (see FIG. 2). For example, the binary code can be stored on a memory card 6, which can later be inserted into a memory card interface 7 of the controller 5.

[0024] The controller 5 can interpret and execute the actual control commands of the sequence 3 of control commands. As mentioned above, the control commands are typically embodied in ASCII code. Conversely, the binary code into which the sequence 3 of control commands is automatically coded by the compressor 1 and the coder 2, is not an ASCII code, as discussed above.

[0025] The additional information 4 can also be compressed and encrypted. According to FIG. 1, the additional information 4 can include a usage limitation NS, an individualization code IC, a user code BC, a comment K as well as a secret key GS. The significance of the additional information NS, IC, BC, K, GS will be described in more detail below with reference to FIG. 2.

[0026] The binary code is compressed relative to the sequence of control commands as a result of the compression in compressor 1. The binary code is also encrypted compared to the sequence 3 of control commands as a result of the encryption in coder 2.

[0027] Neither compression nor encryption reduce the information content. The binary code therefore contains the full information content of the sequence 3 of control commands as well as of the additional information 4. Both the original sequence 3 of control commands as well as the additional information 4 can hence be reconstructed from the binary code.

[0028] In the embodiment of FIG. 1, compression is performed first, followed by encryption. Alternatively, the sequence 3 of control commands could also be only compressed or only encrypted, but not both. This is indicated in FIG. 1 by the dotted line. In principle, compression and encryption could also be performed in the reverse order or in a single step.

[0029] If encryption is omitted, then the additional information 4 becomes useless, with the exception of the comment K. The additional information 4 could then be transformed by a simple decompression back into ASCII code, which people can easily read and understand. On the other hand, a higher compression ratio can be obtained by using only compression (i.e. no encryption), in particular, when the compression algorithm is especially adapted

to the sequence of control commands to be compressed. For example, an encoding unit can be represented by a program set.

[0030] As depicted in FIG. 2, the controller 5 controls a machine tool 8. The controller 5 is implemented herein as a CNC controller.

[0031] The binary code is stored in the controller 5, in the present example by inserting the memory card 6 into the memory card interface 7 of the controller 5.

[0032] When the controller 5 is turned on, a processor 9 initially executes a computer program 10. When executing the computer program 10, the processor 9 initially requests from a user a personal key via a man-machine interface 11. Alternatively, the processor 9 can request this key each time the controller 5 is turned on. The key can also be requested only during the initial startup, whereafter the key is stored in a persistent memory 12 of the processor 9, for example in an EEPROM memory 12.

[0033] The processor 9 transmits the entered personal key to a decoder 13. The decoder 13 then reads the binary code from the memory card 6 and decrypts the binary code using the personal key. After decryption, the sequence 3 of commands and the additional information 4 are obtained by decompression in a decompressor 14 and transmitted to the processor 9.

[0034] During further execution of the computer program 10, the processor 9 recalls, for example, from a timer 15 the absolute time and compares the absolute time with the usage limitation NS. Depending on the comparison, the processor 9 then continues the execution of the computer program 10 or terminates its operation. The number of control commands that can be used from the sequence 3 of control commands can be limited with the usage limitation NS.

[0035] During further execution of the computer program 10, the processor 9 encrypts, for example, the absolute time with the secret key GS and writes the time to the memory card 6 by accessing the memory card interface 7. In this way, for example, any manipulation of the timer 15 by the user of the controller 5 can be identified. This approach can in principle also be applied to other information.

[0036] The processor 9 then queries a processor-internal memory 16 and compares its content with the individualization code IC. Further execution of the computer program 10 is halted or continued depending on the result of this comparison. Processing of sequence 3 of control commands can thereby be restricted by the individualization code to a specific controller 5.

[0037] The memory card 6 can also include a blank code as original individualization code IC. When the processor 9 notices this, it encrypts, for

example, a combination of the current time and the code stored in the processor-internal memory 16 and stores this combination in encrypted form, i.e., encrypted with the secret key GS, in the memory card 6 as a new individualization code IC. It is hence possible to supply a blank memory card 6, so that the initial use of the memory card 6 identifies a specific controller 5 permitted to execute the sequence 3 of control commands. Essentially the same approach can be applied to the user code BC.

[0038] Thereafter, still during execution of the computer program 10, the comment K is outputted via the man-machine interface 11.

[0039] After the computer program 10 has been executed to this point, the execution of the sequence 3 of control commands starts. In particular, the portion of the binary code which includes the sequence 3 of control commands is now decrypted and decompressed. The original sequence 3 of control commands is then reconstructed by the controller from the binary code. The sequence 3 of control commands, however, is not persistently stored in the controller 5. It is either not stored at all or it is stored in a volatile memory 17, preferably by storing only certain segments and by storing these segments dynamically. Immediately after the control commands have been determined, the controller 5 immediately controls the controlled device 8, in this case the machine tool 8. Consequently, the binary code is coded into the sequence 3 of control commands online only during the actual operation of the controller 5.

[0040] According to the diagram of FIG. 2, the decoder 16 and the decompressor 14 are circuit blocks. However, the circuit blocks 13, 14 can also be implemented in software as part of the computer program 10.

[0041] FIG. 2 also shows a configuration where decryption is performed first followed by decompression. In the analogy to FIG. 1, only one of these two measures may be executed. Also, the order in which the measures are executed may be reversed, and the measures may be performed as a single step.

[0042] The present invention provides a number of advantages. In particular, there is no need to change existing methods for generating the sequence 3 of control commands and for processing the sequence 3 of control commands. Only the binary code needs to be generated, based on the sequence 3 of control commands, and the sequence 3 of control commands has to be reconstructed from the binary code. The present invention significantly reduces transmission times and memory requirements in the controller 5.

[0043] It should also be mentioned that the present invention can be applied not only to a complete sequence 3 of control commands, but optionally also to segments of such sequences (subprograms). Moreover, the present invention is not limited to applications in CNC controllers 5. The programmable industrial controller 5 can also be, for example, a memory-programmable controller 5.

[0044] While the invention has been illustrated and described in connection with currently preferred embodiments shown and described in detail, it is not intended to be limited to the details shown since various modifications and structural changes may be made without departing in any way from the spirit of the present invention. The embodiments were chosen and described in order to best explain the principles of the invention and practical application to thereby enable a person skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated.

[0045] What is claimed as new and desired to be protected by Letters Patent is set forth in the appended claims and their equivalents: